Installation on VSE/ESA

This document describes procedures for installing SMARTS under VSE/ESA.

Software AG recommends that you keep unmodified copies of all materials distributed or created as part of the installation process. This may assist with problem diagnosis later.

This document covers the following topics:

- The Installation Tape
- Installing the SMARTS Server Environment
- Installing the VSE CICS Client Environment
- Where Next ?

The Installation Tape

The installation tape is described under the following headings:

- Tape Contents
- Copying Contents of the Tape to Disk

Tape Contents

The installation tape contains the following files:

Dataset	Contains . . .	
APSvrs.LIBR	Library Sublibrary	SAGLIB APSvrs SMARTS components: phases, objects, JCL, sample source members, and macros.

Sample JCL and Source Members

The following table lists the sample source and job members in the APSvrs sublibrary. These must be modified before being used:

Member	Contains ..
APSSIP.J	Sample job to initialize the SMARTS system adapter.
PXANCONF.P	The POSIX server configuration parameters.
PXANHOST.P	Sample parameter file to customize the TCP/IP host name and host address table.
RJANPARM.P	Sample server environment parameters.
RJBNINS1.J	Sample job to restore the SMARTS libraries.
RJBNINS2.J	Sample job to allocate the SMARTS VSAM Dump file.
RJBNINS3.J	Sample job to allocate and restore the MSHP History file.
RJBNINS4.J	Sample job to allocate the SMARTS VSAM Trace file.
RJBNPROC.J	Sample procedure to run the SMARTS Server Environment.

Copying Contents of the Tape to Disk

Step 1: Restore the SMARTS library

- Use the following JCL, supplied in the APSvrs sublibrary as member RJBNINS1.J, to restore the SMARTS library:

```
* $$ JOB JNM=APSREST,CLASS=c,DISP=d,LDEST=(,uid)
* $$ LST CLASS=c,DISP=d
// JOB APSREST  --- Restore APS Library ---
/*
/* ================================================================ *
/* Restore APS Library                                              *
/* ================================================================ *
/*
// PAUSE
// ASSGN  SYS006,cuu
/*
// DLBL   SAGLIB,'saglib.library',0,SD
// EXTENT ,vvvvvv,1,0,ssss,ttt
/*
// MTC    REW,SYS006
// MTC    FSF,SYS006,nn
/*
// EXEC   LIBR
   RESTOR SUB=SAGLIB.APSvrs   : SAGLIB.APSvrs   -
          R=Y TAPE=SYS006
/*
/&
$$ EOJ
```

Installing the SMARTS Server Environment

Step 1: Installing the SMARTS Server Environment

- Use the sample JCL member APSSIP.J in the APSvrs sublibrary to initialize the SMARTS system adapter. Customize the various parameters to suit your needs.

You must execute this JCL before you execute the SMARTS server to avoid initialization errors.

Software AG recommends that you add this JCL to the $ASIPROC so that the SMARTS system adapter is initialized automatically at IPL time.

Step 2: Allocate the SMARTS VSAM Dump file

- Use the sample JCL member RJBNINS2.J in the APSvrs sublibrary to allocate and restore the SMARTS VSAM Dump file. Customize the various parameters to suit your needs.

  The file allocated in this step will be assigned in the SMARTS server start-up JCL.

Step 3: Allocate the SMARTS Trace file

- Allocate either an SD or VSAM/ESDS file for the SMARTS trace file. The APSvrs sublibrary contains a sample JCL to allocate the SMARTS Trace file as a VSAM/ESDS file (member RJBNINS4.J. Customize the various parameters to suit your needs.

  The file allocated in this step will be assigned in the SMARTS server start-up JCL.

Step 4: Allocate the SMARTS History file

- Use the sample member RJBNINS3.J in the APSvrs sublibrary to allocate and restore the MSHP History file. Customize the various parameters to suit your needs.

  This file will be required when applying maintenance to SMARTS.

Step 5: Customize the SMARTS TCP/IP host name and address table

1. Because the current TCP/IP stack on VSE/ESA does not support host name/host address lookup (DNS), SMARTS uses a local address table that mimics the DNS functionality.

   Use the sample host name parameter member PXANHOST.P in the APSvrs sublibrary and customize to suit your needs. When customizing the local table, define:

   - any host names and addresses that will be accessed from within the SMARTS server partition and
   - the host where the local SMARTS server is executing.

   For example, for a local host with name LOCAL and IP address 127.0.0.1 and a remote host with name REMOTE and IP address 255.89.65.90:

   ```
   127.0.0.1      LOCAL  AF_INET
   255.89.65.90 REMOTE AF_INET
   ```

2. Verify and if necessary add or modify the following parameter in the members RJBNPROC.J and PXANCONF.P to point to the PXANHOST.P member:

   ```
   HOSTS_FILE=/SAGLIB/APSvrs/PXANHOST.P
   ```

Step 6: Edit the SMARTS Server start-up JCL

- Modify the sample SMARTS server start-up JCL member RJBNPROC.J in the APSvrs sublibrary to suit your installation naming conventions.

  The example SMARTS start-up JCL below is typical for a VSE/ESA environment and serves as the basis for the various descriptions and explanations that follow:

```
* $$ JOB JNM=RJBNPROC,CLASS=c,DISP=d,LDEST=(,uid)
* $$ LST CLASS=c,DISP=d
// JOB RJBNPROC  --- SMARTS Startup ---
/*
/* ================================================================ *
/* SMARTS Startup                                                   *
/* ================================================================ *
/*
// OPTION PARTDUMP,NOSYSDMP,LOG
/*
/* Dump file for APS ----------------------------------
/*
// DLBL   COMDMP,'aps.vsam.dumpfile',,VSAM,CAT=ccccccc  Step 2
/*
/* Tracing and logging --------------------------------
/*
// ASSGN  SYSnnn,DISK,VOL=vvvvvv,SHR
// DLBL   APSTRCE,'aps.trace.file',0,SD                 Step 3
// EXTENT SYSnnn,vvvvvv,1,0,ssss,ttt
/*
/* Libdefs --------------------------------------------
/*
// LIBDEF PHASE,SEARCH=(SAGLIB.APSvrs,                                +
              SAGLIB.WALvrs)
/*
// UPSI   00000000
// EXEC   TLINSP,SIZE=AUTO
*
* ------------------------------------------------------
* Example SYSPARMS for the SMARTS SERVER Environment (SSE) (RJANPARM.P)
* ------------------------------------------------------
*
 INSTALLATION=SMARTS                  Installation ID
 THREAD-GROUP=(DEFAULT,($DEFAULT,20,2,0,0,N))
 WORKLOAD-MAXIMUM=050
*
 SERVER=(OPERATOR,TLINOPER,TLSPOPER)   Operator Communications Server
 SERVER=(POSIX,PAENKERN)               POSIX                   Server
*
* ------------------------------------------------------
* Example SYSPARMS for the SMARTS POSIX  Environment (PSX) (PXANCONF.P)
* ------------------------------------------------------
*
 ENVIRONMENT_VARIABLES=/SAGLIB/APSvrs/ENVVARS.P
 HOSTS_FILE=/SAGLIB/APSvrs/PXANHOST.P                     Step 5
 LOG=OPER                             Messages to Operator Console
 SYSTEM_ID=SMARTS                     System ID
*
/*
// EXEC   LISTLOG
/*
/&
* $$ EOJ
```

For a description of the SMARTS SYSPARMS, see *Configuration Parameters*. For parameters relevant to your application, refer to the configuration documentation for the software that runs on SMARTS.

Installing the VSE CICS Client Environment

Step 1: Modify the CICS Procedure

1. Add the `APSvrs` sublibrary to the LIBDEF search chain. This must be placed before any other Software AG product sublibrary.
2. Add `DLBL` and `EXTENT` statements for STDOUT and STDERR if using.
3. Add `DLBL` and `EXTENT` statements for APSTRCE.
4. Declare the location of the `SYSPARM` dataset via the `PARM=` option of the `EXEC` statement. For example, if the system parameters are to be found in a sublibrary member the statement could look like this:

   ```
   // EXEC DFHSIP,PARM='SYSPARM(/PROD/CICS/SYSPARM.P)',.....
   ```

5. Add `// OPTION SYSPARM='nn'` statement where nn is the id of the *TCP/IP* for VSE job to be used with CICS. The default is `00` if omitted.

Step 2: Define Transactions to CICS

Four transactions are required: `TDSP`, `SMGO`, `SMEX`, `SMNE`. These transactions must be defined as follows:

1. `SMGO` to run `PACNKERN`. The name `SMGO` is mandatory as it is used internally by SMARTS.
2. `SMEX` to run `PACNKERX`. `SMEX` is a suggested transaction name.
3. `TDSP` to run `PACNSTRT`. The name `TDSP` is mandatory as it is used internally by SMARTS.
4. `SMNE` to run `PACNNEP`. The name `SMNE` is mandatory, as it is used internally by SMARTS.

   A sample job is provided to add these transactions to the CSD.

Step 3: Define Programs to CICS

- Many programs are used by SMARTS so autoinstall should be activated. Autoinstall can be activated by setting the `SIT` parameter `PGAIPGM=ACTIVE`

  The following Assembler language programs must be defined to CICS.

  - `PACNKERN`, define with `EXECKEY(CICS)`
  - `PACNKERX`, define with `EXECKEY(CICS)`
  - `PACNSTRT`, define with `EXECKEY(CICS)`

- PACNNEP, define with EXECKEY(CICS)
- RAANPARM, define with "RELOAD=YES"

A sample job is provided to add these programs to the CSD.

Step 4: Define Programs to the Program List Tables

1. To automatically initialise SMARTS POSIX at CICS startup, in the DFHPLTxx for the PLTPI, insert PACNSMGO as a second phase PLT program.
2. To automatically terminate SMARTS POSIX at CICS shutdown, in the DFHPLTxx for the PLTSD, insert PACNKERX as a first phase PLT program.

Step 5: Provide a DFHZNEP node error program

- If the installation already has a DFHZNEP node error program in use, modify it to invoke the SMNE transaction under the conditions detailed in the model assembler program PACNZNEP, supplied in the APSvrs sublibrary. If the installation does not have a DFHZNEP node error program in use, use the supplied model program PACNZNEP to create one.

Step 6: Define transaction security

- Each of the four SMARTS transactions may be defined with only basic security to the security manager installed. SMGO and SMEX are the only transactions that may be entered at a terminal and may be protected as required.

Where Next ?

You have now installed the SMARTS software. You can continue now with the installation of the application that is to run on SMARTS.

Note that the configuration procedure of the application that runs of SMARTS may instruct you to modify some of SMARTS’s configuration parameters.